第五周神经网络-反向传播算法

1. 神经网络的代价函数

L表示网络一共有几层。

K表示输出层的神经元个数。

M代表样本数据的个数。

代价函数如下：（与逻辑回归代价函数类似）



1. 反向传播算法

为了最小化代价函数，使用反向传播算法。

目标：

算法解决步骤：

1. 让
2. 正向传播计算
3. 用计算输出层的误差
4. 反向传播算法计算，
5. 得出，向量化有以下：





可用来实现目标。

实际有：





可以计算：



展开参数：

为了使用优化方法fminunc()，将theta和delte展开成长向量形式，操作如下：

ThetaValue=[theta1(:);theta2(:);theta3(:)];

Delte=[delte1(:);delte2(:);delte(3)];

得到thetavalue和delte都为长向量形式。

也可以通过：

Theta1=reshape(thetavalue(1:n),i,j);这样的形式重新获得theta矩阵。

梯度检查：

当使用反向传播算法计算DeltaVector时，可以通过以下方法进行检验：

首先需要使theta矩阵长向量化，然后有：



一般要求epsilon很小，但不能太小，太小会有计算问题。

Octave操作如下：

Epsilon=1e-4;

For i=1:n,

Theta1=theta;

Theta1(i)=theta1(i)+epsilon;

Theta2=theta;

Theta2(i)=theta2(i)-epsilon;

Grad(i)=(J(theta1(i))-J(theta2(i)))/(2\*epsilon);

最后得到grad，若deltavectorgrad，则可证明反向传播算法的正确性。当证明正确性后，则不需要进行梯度检查了。

参数随机初始化：

如果初始化参数为0,则梯度下降时会出现更新的值为相同的值，降低了算法性能，所以采用随机参数初始化。操作如下：

 

神经网络学习算法:

1. 首先选择一个网络框架，确定隐藏层层数，默认为1层。输入层根据特征矩阵来确定。输出层根据类别种类来确定。
2. 随机初始化权重参数theta。
3. 输入特征值，进行向前传播，得到。
4. 实现并计算代价函数。
5. 根据输出值进行反向传播，得到。
6. 用梯度检查反向传播得到的deltavector与数值计算得到的grad是否相近来确定反向传播是否正确。
7. 取消梯度检查，使用得到的deltavector最小化。

注意：神经网络代价函数不是下凸函数，所以最后结果是局部最优，与初始参数有关。

编程作业：

代码主要功能是计算代价函数、实现向前传播计算预测输出值、实现向后传播计算delta，从而优化参数theta（即达到训练效果）。

%向前传播X=[ones(m,1),X]; %5000\*401hidden1\_net=X\*Theta1'; %5000\*25hidden1\_out=sigmoid(hidden1\_net); %5000\*25hidden1\_out=[ones(m,1),hidden1\_out]; %5000\*26hidden2\_net=hidden1\_out\*Theta2'; %5000\*10hidden2\_out=sigmoid(hidden2\_net); %5000\*10for i=1:m labels=zeros(num\_labels,1);%初始化labels为10\*1的矩阵 result=y(i);%拿出结果 labels(result)=1;%结果位置设置为1 J=J+log(hidden2\_out(i,:))\*(-labels)-log(1-hidden2\_out(i,:))\*(1-labels);%累加代价函数 %向后传播 delta2 = hidden2\_out(i, :)' - labels; %计算输出层差值 delta1 = Theta2(:, 2:end)' \* delta2 .\* sigmoidGradient(hidden1\_net(i, :)');%计算隐藏层的差值

Theta2\_grad = Theta2\_grad + delta2 \* hidden1\_out(i, :); %优化参数

Theta1\_grad = Theta1\_grad + delta1 \* X(i, :);end

sigmoidGradient函数：

g=sigmoid(z).\*(1-sigmoid(z)); % g(z)’的导数，sigmodi的导数

学习总结：

本周学习了神经网络的代价函数、向后传播算法以及算法检验，通过后向传播算法，计算各单元的误差delta值，不断更新theta值，使输出结果不断接近y值。这次的编程作业在实现中遇到了困难，通过查阅相关资料得以解决并理解。前两周前向传播和后向传播能够有效的实现预测，两种传播相辅相成，相互依赖。

后向传播公式证明参考：

题目：反向传播算法的直观理解

http://blog.csdn.net/mao\_xiao\_feng/article/details/53048213